米国におけるソフトウェア高信頼化の最新動向

～カーネギーメロン大学ソフトウェア・エンジニアリング研究所幹部による特別セミナー講演より～

SEC ソフトウェアグループ リーダー
中尾　昌善

1.　はじめに

カーネギーメロン大学ソフトウェア・エンジニアリング研究所（通称：SEI）は、私ども SEC 設立時のお手本となった組織であり、米国における最先端のソフトウェア工学の推進を行っている研究所である。この度、7 月 11 日に SEC 特別セミナーを開催し、同研究所の Paul D.Nielsen 所長と James W.Over テクニカルディレクターにご講演いただいた。本稿では、両氏の講演の概要を報告すると共に、所感を記述させていただく。

2.　Paul D.Nielsen 所長の講演概要

多くのセキュリティ問題事例

ある調査によれば、69％ものソフトウェアで、トップ 25 の危険に分類されるような脆弱性につながるコーディングがなされている。例を幾つか示すと、OpenSSL の脆弱性バグは 2 年間見つからずに放置され、全世界に影響を与えた。あるオンラインバンキングでは、パスワードとユーザ名が盗まれ、２億円相当が盗まれた。また、大手デパートでは、内部のアクセス権限を持つ者が取り込まれ、4,000 万件のパスワードが盗まれるという事件も起きた。これは、評判の失墜につながったのが致命的であった。Netflix のアプリケーションには、内部犯罪によりマルウェアが埋め込まれており、インストールすると、勝手にロシアにつながり、クレジット情報が盗まれてしまった。既に、多数の組込み型製品がネットワークにつながる時代になっており、脅威は膨大なものとなっている。

開発段階から考慮すべきセキュリティ対策

サイバーセキュリティに対応するには、開発の早い段階でセキュリティ要件を考える必要がある。後で追加的に考えるのはかなり難しい。つまり、設計時にセキュリティ要件を規定し、それを担保するアーキテクチャを考えていく必要がある。更に、脆弱性につながる欠陥は、ソフトウェアの構築時に早期に見つけるだけでなく、以後の追加変更時にも完全性を追求していく必要がある。

ソフトウェアのコーディングによる問題は多いが、その大半は予防可能である。SEI では過去 15 年間にわたって、25 個の主要なセキュリティ問題への対応を考えてきた結果、回避策をセキュアコーディングスタンダードとして標準化し、コンパイラでの自動チェックに結びつけている。

また、アシュアランスケースの活用により、品質保証とミッション保証に対応してきた。

アーキテクチャ設計の重要性

アジャイル開発は合理的手法だが、アーキテクチャをあまり考慮しない。小規模システムでは構わないが、大規模システムではアーキテクチャを最初から考えておかないと、手戻りが発生する。シンプルな構成にすることは必要だが、ある程度のアーキテクチャは最初から考えておくことが必要である。

経営者に求められるソフトウェア開発への理解

ここ 10 年に発展してきた企業を見ると、FedEx やアマゾンに限らず、ソフトウェアが事業推進の中核になっている企業が多い。そこでは、エンジニアだけでなく、経営陣においても、ソフトウェアを理解し、それが経営に及ぼすリスクを把握し、対応を考えていくことが重要である。また、防止だけでなく、たとえ攻撃されたとしても、ダメージを極力抑え、復旧コストを最小限に抑える取り組みも必要である。

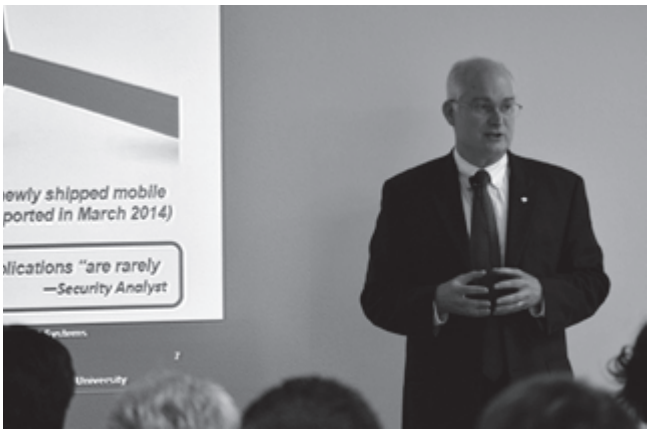

写真１　Paul D.Nielsen 所長

今後の課題

インサイダー事件が、今後の課題である。国家スパイや産業スパイ、会社への不満等による事業妨害、仲間の口座への振り込み不正等が主な問題として存在する。

ネットワークでつながり、複雑化するシステムでは、このような問題への単独での対応には限界があるため、世界中の国や企業が協力して、取り組んでいく必要がある。

3. James W.Over テクニカルディレクターの講演概要

「神ならば信じよう。そうでなければ、データを持ってきなさい。」

上記は、Deming 博士の有名な言葉である。開発プロジェクトにおいては、直感に頼ることも可能だが、その直感に合わないものを見過ごしてしまう危険がある。より良い開発マネジメントを行うには、開発プロセスから出てくるデータをきちんと捕捉し、評価・分析することが重要である。

開発データの測定フレームワーク

しかし、ソフトウェアにおいては、収集するデータの定義が未確立で、精度も高くないのが現実であった。例えば、「タスク時間」には、誰のどういう活動を含めるのかということがあいまいであった。そこで、TSP（Team Software Process）では、測定フレームワークを作った。それは、①成果物のサイズ（行数、機能数等）、②開発者がタスクに実際に使った時間、③欠陥（バグだけでなく、文書等も含めて修正が必要になるものすべて）、④リソースの割り当て、⑤スケジュール　である。週に 1 回はデータを見て検証し、プロジェクトの節目節目にも評価・分析を行う。ここで集めたデータをデータベース化し、それを参照できるようにしている。

パフォーマンス評価手順

プロジェクトのパフォーマンス評価は、次のような手順で行っている。

（1）データの取り込み　データのバックグラウンドチェックや一貫性チェックを行い、データの正当性を確認する。

（2）統計分析　データには過去データに基づく分布パターンがあるので、それに対するハズレ値がないかを確認する。原因がデータの不正確性にあると判明すれば、データから除外する。こうして、ベースラインデータが得られる。

（3）ファクトシート　ベースラインデータを元に、プロジェクト、プロダクト、プロセスの 3 つの観点のファクトシートができる。

（4）ベンチマーク　他のプロジェクトや企業と比較する。

（5）証明書の発行　当該プロジェクトのレベルを評価した報告書を証明書として各企業に送付する。

（以後は、例を提示しつつ、上記の具体的やり方の解説がなされた。）

興味深い標準データ

これまでに得られている興味深いデータの幾つかを紹介すると、①エンジニアは、1 週間約 40 時間の労働時間のうち、正味のタスク実施時間は平均約 10 時間である。②生産性の平均は、10 行／時間である。

写真 2　James W.Over テクニカルディレクター

4. 所感

Nielsen 所長の講演は、セキュリティ上の問題の根源がソフトウェアの開発段階にまでさかのぼることを示し、そこでの対策の重要性を訴えるものであった。私たちも、セキュリティ問題だけでなく、安全上の問題も同様であると考えており、ソフトウェアの開発段階での対策の必要性には共感を得るものがあった。

Over 氏の講演については、エンジニアが 1 週間に約 10 時間しか正味のタスクに時間をかけていないということに対して、会場から多くの異論や質問が続出した。「正味の」という解釈が難しいが、それにしても直感的に少なすぎるという意見が多かった。まさしく、データがあればこそ議論が生じる事例であり、まずはデータを収集することの必要性を図らずも示すことになった。

5. おわりに

このセミナーの開催日は、折しも過去最大級と言われた台風がセミナー会場のある関東に接近する日と重なってしまった。開催が危ぶまれたが、台風は関東の南方を通過し、ほとんど交通にも影響なく平穏な状態で開催することができた。両氏の講演に対して、活発な質問が相次ぎ、参加いただいた方の関心の高さが伺えた。IPA は、今後も SEI との情報交換を通じて、ソフトウェア・エンジニアリングの最新動向を発信していきたいと考えている。